



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/607,560	06/29/2000	Shekhar N. Swamy	MS147163.1	8517
27195	7590	09/01/2004	EXAMINER	
AMIN & TUROCY, LLP 24TH FLOOR, NATIONAL CITY CENTER 1900 EAST NINTH STREET CLEVELAND, OH 44114			KISS, ERIC B	
			ART UNIT	PAPER NUMBER
			2122	

DATE MAILED: 09/01/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary

Application No.

09/607,560

Applicant(s)

SWAMY ET AL.

Examiner

Eric B. Kiss

Art Unit

2122

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
 - If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
 - If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
 - Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 07 June 2004.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-24 and 26-30 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1, 2, 16, 18, 20, 21, 23 and 26-30 is/are rejected.
- 7) ☒ Claim(s) 3-15, 17, 19, 22 and 24 is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 29 June 2000 is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
- Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
- Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
2. ☐ Certified copies of the priority documents have been received in Application No. _____.
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
- * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- 1) ☐ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date _____
- 4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____
- 5) ☐ Notice of Informal Patent Application (PTO-152)
- 6) ☐ Other: _____

DETAILED ACTION

1. The reply filed 7 June 2004 has been received and entered. Claims 1-24 and 26-30 are pending.

Response to Amendment

2. Applicant's amendments to claims 1, 20, 29, and 30 appropriately address the rejection of claims 1-24 and 26-30 under 35 U.S.C. §112, first paragraph, based on lack of enablement. Accordingly, this rejection is withdrawn in view of Applicant's amendments.

Response to Arguments

3. Applicant's arguments filed 7 June 2004 have been fully considered but they are not persuasive.
 - a. In response to Applicant's arguments on p. 14, as described on page 32, in paragraph 4, and in Algorithm 3.2 on pages 33-34, the production group associations are used to generate target subtrees by processing the corresponding source subtrees. Thus, the Examiner maintains that a tracing through the mapping from target to source is necessary in order to process source subtrees corresponding to the target subtrees.

b. In response to Applicant's arguments on p. 15, in the second paragraph, the Examiner respectfully submits that the allocation run-time memory discussed in the cited section of Aho et al. is performed at compile time. Aho et al. teach that is beneficial to statically allocate as many data objects as possible so that the addresses of these objects can be compiled into the target code (p. 396, second-to-last paragraph).

c. In response to Applicant's arguments on p. 15, the Examiner maintains that Bellina teaches that it has been known to generate XSL style sheet representations of mappings from a source schema (an XML file) to a target schema ("another format", for example, an HTML file; see "**XSL generator**" on p. 12).

4. In response to Applicant's unpersuasive arguments, the previous rejections under 35 U.S.C. §§ 102(b), 103(a), are maintained and reproduced below.

Claim Rejections - 35 USC § 102

5. The text of those sections of Title 35, U.S. Code not included in this action can be found in a prior Office action.

Art Unit: 2122

6. Claims 1, 20, 21, 23, 26, 29, and 30 are rejected under 35 U.S.C. 102(b) as being anticipated by Greger Lindén, "Structured Document Transformations," 1997, University of Helsinki, Finland, Series of Publications A, Report A-1997-2 (hereinafter *Lindén*).

As per claim 1, *Lindén* discloses a method of generating code representing a mapping between a source schema and a target schema, the mapping comprising defined (see, for example, page 43, paragraph 1, "ALCHEMIST also provides a graphical interface for specifying transformations") data transformations from the source schema to the target schema, the source schema comprising a source tree having a source node and the target schema comprising a target tree having a target node (see "TT-grammars," description on pages 31-34 and illustrative example on pages 34-42), the method comprising: determining source node dependencies for the target node by tracing from the target node through the mapping to the source schema (specifying production group associations and symbol associations; see pages 49-51; and page 32, second paragraph). As disclosed by Lindén (see, for example, pages 49-51), a mapping connects the source and target grammars together based on the TT-grammar technique. In generating the mapping, the user connects source and target subgrammars by forming production group associations and specifying symbol associations. As described on page 32, in paragraph 4, and in Algorithm 3.2 on pages 33-34, the production group associations are used to generate target subtrees by processing the corresponding source subtrees. Thus, a tracing through the mapping from target to source is necessary in order to process source subtrees corresponding to the target subtrees.

Lindén further discloses matching hierarchy by generating a hierarchy match list for the target node (constructing the corresponding target parse tree; see “source-to-target mapper” description on page 56). As disclosed by *Lindén* (see, for example, page 56), a target parse tree corresponding to the source parse tree is constructed according to the TT-grammar. The target parse tree is a hierarchical data structure containing target nodes that are matched (corresponding to) the hierarchical data structure of source parse tree nodes based on the TT-grammar and corresponding specified mapping.

Lindén further discloses generating code according to the hierarchy match list (see subsection 4.3.2 on pages 51-52). As disclosed by *Lindén* (see, for example, section 4.3.2 on pages 51-52), code is generated according to the mapping and target parse tree. *Lindén* discloses specialized tools, such as STONE, which generates a mapper from the mapping specification, and SWINDLER, which generates an unparser from the target grammar.

As per claim 20, *Lindén* discloses a method for compiling a defined (see, for example, page 43, paragraph 1, “ALCHEMIST also provides a graphical interface for specifying transformations”) mapping between a source schema having source nodes associated therewith, and a target schema having target nodes associated therewith (see “TT-grammars,” description on pages 31-34 and illustrative example on pages 34-42), comprising: determining source node dependencies for at least one target node by tracing from the at least one target node through the mapping to the source schema (specifying production group associations and symbol associations; see pages 49-51; and page 32, second paragraph). As disclosed by *Lindén* (see, for example, pages 49-51), a mapping connects the source and target grammars together based on

the TT-grammar technique. In generating the mapping, the user connects source and target subgrammars by forming production group associations and specifying symbol associations. As described on page 32, in paragraph 4, and in Algorithm 3.2 on pages 33-34, the production group associations are used to generate target subtrees by processing the corresponding source subtrees. Thus, a tracing through the mapping from target to source is necessary in order to process source subtrees corresponding to the target subtrees.

Lindén further discloses matching hierarchy by generating a hierarchy match list for the target node (constructing the corresponding target parse tree; see “source-to-target mapper” description on page 56). As disclosed by *Lindén* (see, for example, page 56), a target parse tree corresponding to the source parse tree is constructed according to the TT-grammar. The target parse tree is a hierarchical data structure containing target nodes that are matched (corresponding to) the hierarchical data structure of source parse tree nodes based on the TT-grammar and corresponding specified mapping.

Lindén further discloses generating code according to the hierarchy match list (see subsection 4.3.2 on pages 51-52). As disclosed by *Lindén* (see, for example, section 4.3.2 on pages 51-52), code is generated according to the mapping and target parse tree. *Lindén* discloses specialized tools, such as STONE, which generates a mapper from the mapping specification, and SWINDLER, which generates an unparser from the target grammar.

As per claims 21 and 23, *Lindén* further discloses generating a source dependency list (specifying production group associations and symbol associations; see pages 49-51; and page

32, second paragraph); and initializing node dependencies memory prior to determining source dependencies and later freeing that memory (inherent).

As per claim 26, *Lindén* further discloses matching hierarchy comprising top-down matching (see Algorithm 3.1 on pages 29-30).

As per claim 29, *Lindén* discloses a system for generating code representing a defined (see, for example, page 43, paragraph 1, “ALCHEMIST also provides a graphical interface for specifying transformations”) mapping between a source schema and a target schema, the mapping comprising data transformations from the source schema to the target schema, the source schema comprising a source tree having source nodes and the target schema comprising a target tree having a target node (see “TT-grammars,” description on pages 31-34 and illustrative example on pages 34-42), the system comprising: means for determining source node dependencies for the target node by tracing from the target node through the mapping to the source schema (specifying production group associations and symbol associations; see pages 49-51; and page 32, second paragraph). As disclosed by *Lindén* (see, for example, pages 49-51), a mapping connects the source and target grammars together based on the TT-grammar technique. In generating the mapping, the user connects source and target subgrammars by forming production group associations and specifying symbol associations. As described on page 32, in paragraph 4, and in Algorithm 3.2 on pages 33-34, the production group associations are used to generate target subtrees by processing the corresponding source subtrees. Thus, a tracing through the mapping from target to source is necessary in order to process source subtrees corresponding to the target subtrees.

Lindén further discloses means for matching hierarchy by generating a hierarchy match list for the target node (constructing the corresponding target parse tree; see “source-to-target mapper” description on page 56). As disclosed by *Lindén* (see, for example, page 56), a target parse tree corresponding to the source parse tree is constructed according to the TT-grammar. The target parse tree is a hierarchical data structure containing target nodes that are matched (corresponding to) the hierarchical data structure of source parse tree nodes based on the TT-grammar and corresponding specified mapping.

Lindén further discloses means for generating code according to the hierarchy match list (see subsection 4.3.2 on pages 51-52). As disclosed by *Lindén* (see, for example, section 4.3.2 on pages 51-52), code is generated according to the mapping and target parse tree. *Lindén* discloses specialized tools, such as STONE, which generates a mapper from the mapping specification, and SWINDLER, which generates an unparser from the target grammar.

As per claim 30, *Lindén* discloses a computer-readable medium having computer-executable instructions for: generating code representing a defined (see, for example, page 43, paragraph 1, “ALCHEMIST also provides a graphical interface for specifying transformations”) mapping between a source schema and a target schema, the mapping comprising data transformations from the source schema to the target schema, the source schema comprising a source tree having a source node and the target schema comprising a target tree having a target node (see “TT-grammars,” description on pages 31-34 and illustrative example on pages 34-42); determining source node dependencies for the target node by tracing from the target node through the mapping to the source schema (specifying production group associations and symbol

associations; see pages 49-51; and page 32, second paragraph). As disclosed by Lindén (see, for example, pages 49-51), a mapping connects the source and target grammars together based on the TT-grammar technique. In generating the mapping, the user connects source and target subgrammars by forming production group associations and specifying symbol associations. As described on page 32, in paragraph 4, and in Algorithm 3.2 on pages 33-34, the production group associations are used to generate target subtrees by processing the corresponding source subtrees. Thus, a tracing through the mapping from target to source is necessary in order to process source subtrees corresponding to the target subtrees.

Lindén further discloses matching hierarchy by generating a hierarchy match list for the target node (constructing the corresponding target parse tree; see “source-to-target mapper” description on page 56). As disclosed by Lindén (see, for example, page 56), a target parse tree corresponding to the source parse tree is constructed according to the TT-grammar. The target parse tree is a hierarchical data structure containing target nodes that are matched (corresponding to) the hierarchical data structure of source parse tree nodes based on the TT-grammar and corresponding specified mapping.

Lindén further discloses generating code according to the hierarchy match list (see subsection 4.3.2 on pages 51-52). As disclosed by Lindén (see, for example, section 4.3.2 on pages 51-52), code is generated according to the mapping and target parse tree. Lindén discloses specialized tools, such as STONE, which generates a mapper from the mapping specification, and SWINDLER, which generates an unparser from the target grammar.

Claim Rejections - 35 USC § 103

7. The text of those sections of Title 35, U.S. Code not included in this action can be found in a prior Office action.

8. Claims 2 and 16 are rejected under 35 U.S.C. 103(a) as being unpatentable over *Lindén* in view of Alfred V. Aho et al. "Compilers: Principles, Techniques, and Tools," 1986, Addison-Wesley (hereinafter *Aho et al.*).

As per claim 2, *Lindén* discloses such a method (see disclosure applied above to claim 1) but fail to expressly disclose allocating memory for a compiler node; associating the compiler node with the target node; allocating memory for compiler variable classes; and associating compiler variable classes with functoids. However, *Aho et al.* teach allocating memory for a compiler node (obtaining a block of storage); associating the compiler node with the target node (subdividing the storage to hold the generated target code); allocating memory for compiler variable classes (subdividing the storage to hold data objects); and associating compiler variable classes with functoids (keeping track of procedure activations; see "Subdivision of Run-Time Memory" on pages 396-397). Therefore, it would have been obvious to one having ordinary skill in the computer art at the time the invention was made to modify the method of *Lindén* to include memory initialization as per the teachings of *Aho et al.* One would be motivated to do so to establish a run-time environment.

As per claim 16, in addition to the disclosure and teachings applied above, *Lindén* further discloses generating a code header for a root node (pre-processing commands); generating a code

trailer for the root node (post-processing commands); processing target record nodes in a preexecuteparent function, a postexecuteparent function, and an executeleaf function (processing the pre-processing and post-processing commands); and processing field nodes in the executeleaf function (processing normal spell commands; see the last paragraph on page 55 through the beginning of section 4.4 on page 57). Therefore, for reasons stated above, such a claim also would have been obvious.

9. Claims 18 and 28 are rejected under 35 U.S.C. 103(a) as being unpatentable over *Lindén* in view of Alberto Bellina, "XmlTool documentation," 21 January 2003 (hereinafter *Bellina*).

As per claims 18 and 28, *Lindén* discloses such a method (see disclosure applied to claims 1 and 28 above) but fails to expressly disclose creating an XSL style sheet representation of the mapping. However, *Bellina* teaches a tool and method of manipulating XML schemas including generating XSL style sheet representations of mappings (see "XSL generator" on page 12). Therefore, it would have been obvious to one having ordinary skill in the computer art at the time the invention was made to modify the method of *Lindén* to include generation of XSL style sheet representations as per the teachings of *Bellina*. One would be motivated to do so to be able to produce output in a standard transformation language format for use with XML files.

10. Claims 27 is rejected under 35 U.S.C. 103(a) as being unpatentable over *Lindén*.

As per claim 27, *Lindén* discloses such a method (see disclosure applied above to claim 20) and furthermore discloses the transformation of SGML schemas (see section 2.4 on pages 20-23; and introduction to chapter 5 on page 59) but fails to expressly disclose the source schema

Art Unit: 2122

and target schema being XML schemas. However, *Lindén* teaches the introduction of XML as a subset of SGML and suggests its future use as a replacement for SGML (see last paragraph on page 23). Therefore, it would have been obvious to one having ordinary skill in the computer art at the time the invention was made to modify the SGML schema transformation method of Lindén to include XML schemas as per Lindén's own suggestion. One would be motivated to do so to implement schemas that lack the drawbacks of full SGML.

Allowable Subject Matter

11. Claims 3-15, 17, 19, 22, and 24 are objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims.

Conclusion

12. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

13. Any inquiry concerning this communication or earlier communications from the Examiner should be directed to Eric B. Kiss whose telephone number is (703) 305-7737. One or around October 19, 2004, Technology Center 2100 will be relocated to Alexandria, Virginia, and Examiner Kiss's telephone number will change to (571) 272-3699. The Examiner can normally be reached on Tue. - Fri., 7:15 am - 4:45 pm. The Examiner can also be reached on alternate Mondays.

If attempts to reach the Examiner by telephone are unsuccessful, the Examiner's supervisor, Tuan Dam, can be reached on (703) 305-4552. On or around October 19, 2004, Technology Center 2100 will be relocated to Alexandria, Virginia, and Tuan Dam's phone

Art Unit: 2122

number will change to (571) 272-3695. The fax phone number for the organization where this application or proceeding is assigned is (703) 872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

EBK/~~EBK~~
August 25, 2004



ANTONY NGUYEN-BA
PRIMARY EXAMINER